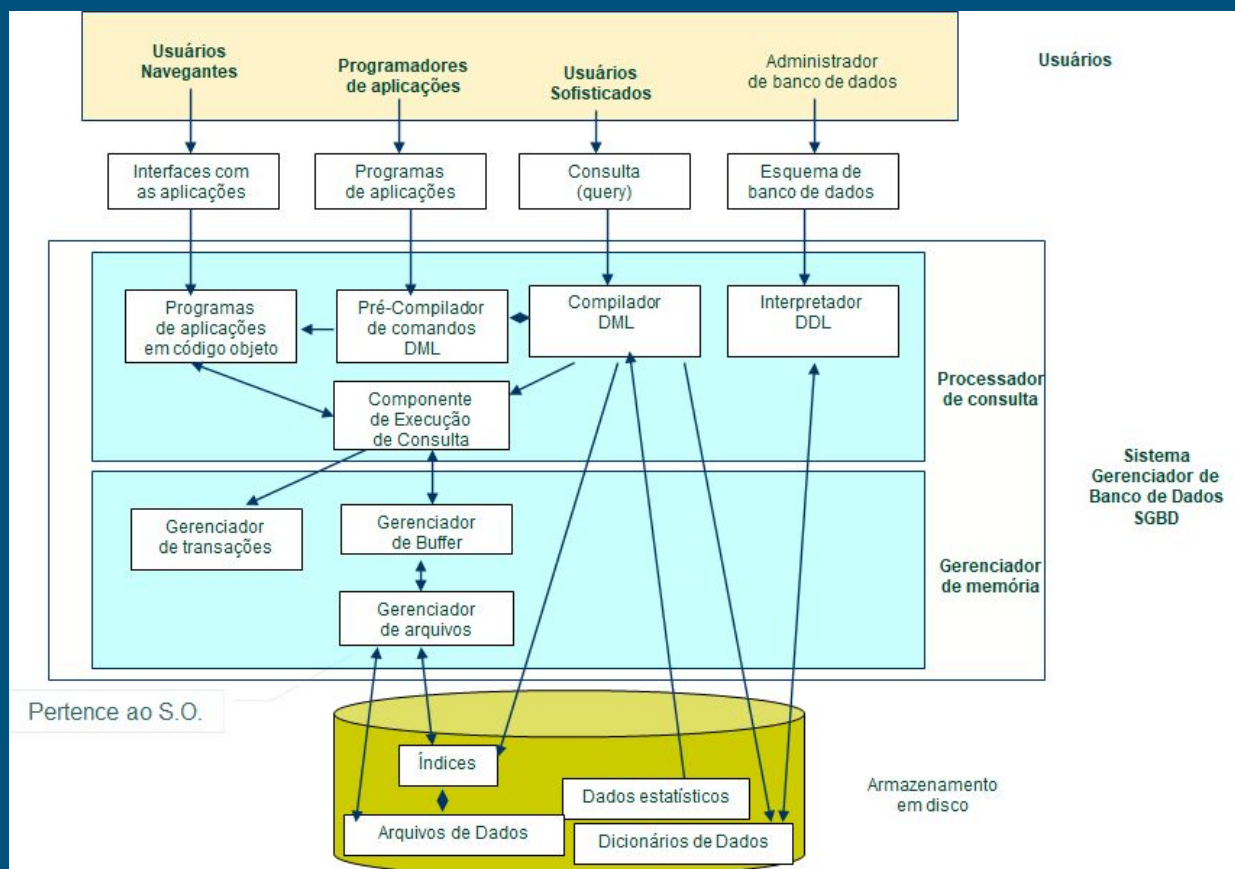


Aula 08

Banco de Dados

emerson@paduan.pro.br



emerson@paduan.pro.br

Modelos de dados - Níveis de abstração

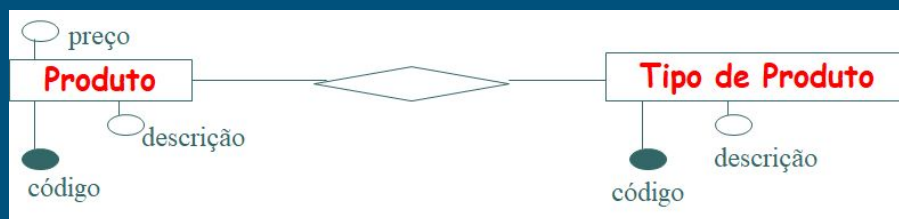


emerson@paduan.pro.br

Modelo Conceitual

Descreve a estrutura de um banco de dados independente de um SGBD

Não define: como esses dados serão armazenados em nível de SGBD



emerson@paduan.pro.br

Modelo Lógico

Representa a organização de dados em um banco de dados de acordo com o SGBD a ser utilizado.

<u>CodTipoProd</u>	<u>DescrTipoProd</u>		
1	Computador		
2	Impressora		

<u>CodProd</u>	<u>DescrProd</u>	<u>PrecoProd</u>	<u>CodTipoProd</u>
1	PC desktop modelo X	2.500	1
2	PC notebook ABC	3.500	1
3	Impressora jato de tinta	600	2
4	Impressora laser	800	2

TipoDeProduto(CodTipoProd, DescrTipoProd)

Produto(CodProd, DescrProd, PrecoProd, CodTipoProd)
CodTipoProd referencia TipoDeProduto

emerson@paduan.pro.br

Modelo Físico

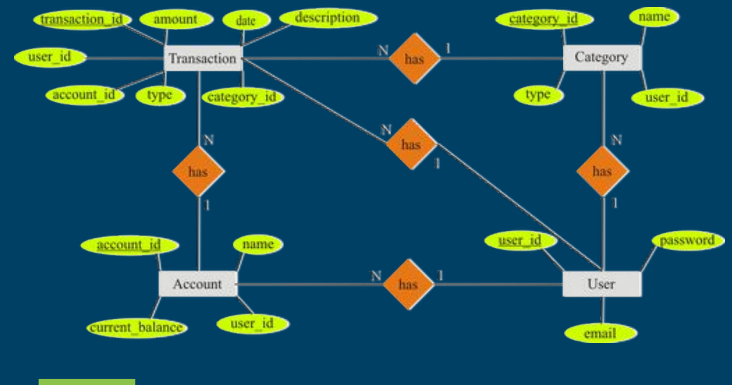
Contém detalhes de armazenamento interno de informações e depende do SGBD a ser utilizado.

Nome da Tabela	PRODUTO			
Descrição	ESTRUTURA QUE ARMAZENA OS DADOS DOS PRODUTOS VENDIDOS NA EMPRESA			
Coluna	Descrição	Tipo de dado	Nulo	Consistência
CodProd	Código do produto	Number(3)	N	PK
Descr_Prod	Descrição do Produto	Varchar(40)	N	
PrecoProd	Preço do produto	Number(10,2)	N	
CodTipoProd	Código do Tipo do Produto	Number(3)	N	FK(TipoDeProduto)

emerson@paduan.pro.br

Modelo Entidade Relacionamento

Diagrama
Entidade-Relacionamento (DER)



Conceitos fundamentais

- Entidade
- Atributo
- Relacionamento
- Cardinalidade
- Generalização/Especialização

Entidade

➤ Entidade

- representa objetos concretos (uma pessoa, um automóvel) ou abstratos (um departamento, um endereço)
- o nome é usado no singular
- no DER representada por um retângulo contendo o nome da entidade

Cliente

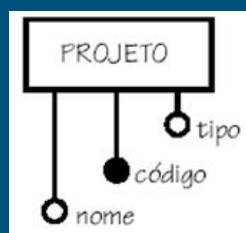
Departamento

emerson@paduan.pro.br

Atributos

➤ Atributos

- são as propriedades sobre as quais temos interesse na Entidade
- ligadas à entidade à qual pertencem

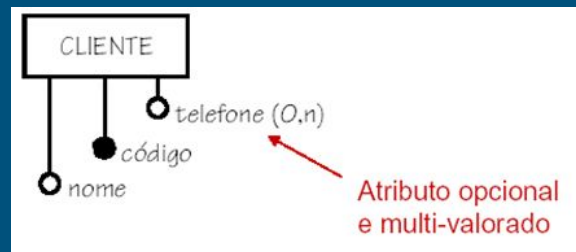


emerson@paduan.pro.br

Atributos

➤ Tipos de atributos

- simples / composto
- univalorado / multivalorado
- identificador
- opcional
- derivado



emerson@paduan.pro.br

Como identificar

➤ Como identificar as entidades?

- A partir de uma descrição, identificar os objetos "participantes" da descrição.

➤ E os atributos?

- Características, informações, que descrevem esses objetos

emerson@paduan.pro.br

Exercício: Como identificar

Deseja-se construir um banco de dados para um sistema de vendas. Em cada venda são vendidos vários produtos e um determinado produto pode aparecer em diferentes vendas. Cada venda é efetuada por um vendedor para um determinado cliente. Um produto está armazenado em uma prateleira.

emerson@paduan.pro.br

Exercício: Como identificar

O objetivo é construir um banco de dados para controlar o aluguel de veículos. A locadora possui várias agências espalhadas pelo país. Os atendentes fazem o aluguel para os clientes. Sabe-se que cada veículo é alugado para um cliente em específico e cada cliente pode alugar vários carros. Sabe-se também que os carros sofrem manutenções periódicas.

emerson@paduan.pro.br

Relacionamento

➤ Relacionamento

- Conjunto de associações entre entidades
- Observe os verbos

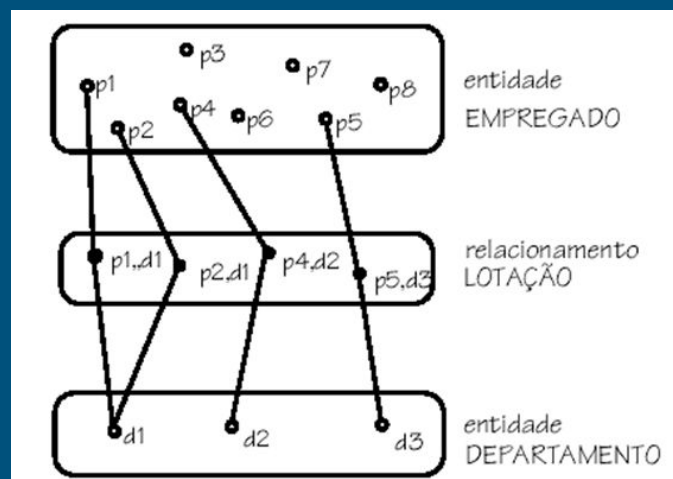


emerson@paduan.pro.br

Relacionamento

➤ Relacionamento

- Diagrama de ocorrências



emerson@paduan.pro.br

Exercício: Como identificar

Deseja-se construir um banco de dados para um sistema de vendas. Em cada venda são vendidos vários produtos e um determinado produto pode aparecer em diferentes vendas. Cada venda é efetuada por um vendedor para um determinado cliente. Um produto está armazenado em uma prateleira.

emerson@paduan.pro.br

Exercício: Como identificar

O objetivo é construir um banco de dados para controlar o aluguel de veículos. A locadora possui várias agências espalhadas pelo país. Os atendentes fazem o aluguel para os clientes. Sabe-se que cada veículo é alugado para um cliente em específico e cada cliente pode alugar vários carros. Sabe-se também que os carros sofrem manutenções periódicas.

emerson@paduan.pro.br

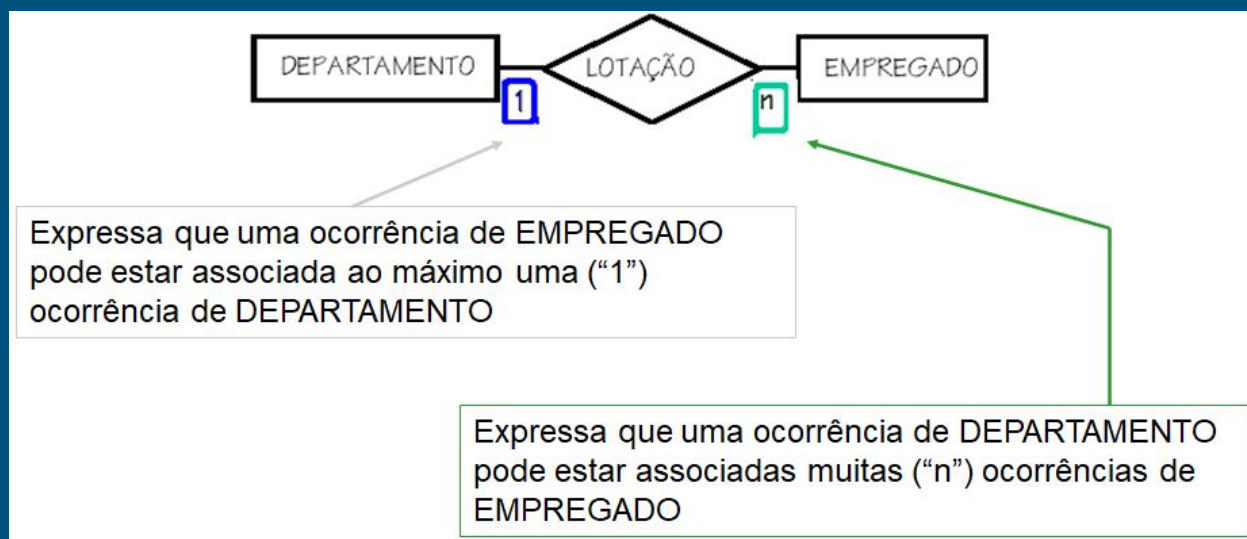
Cardinalidade

> Cardinalidade

- Representa quantas ocorrências de uma entidade podem estar associadas a uma determinada ocorrência de entidade através do relacionamento
- Pode ser usada para classificar relacionamentos binários (2 entidades)
 - $n : n$ (muitos para muitos)
 - $1 : n$ (um para muitos)
 - $1 : 1$ (um para um)

emerson@paduan.pro.br

Leitura



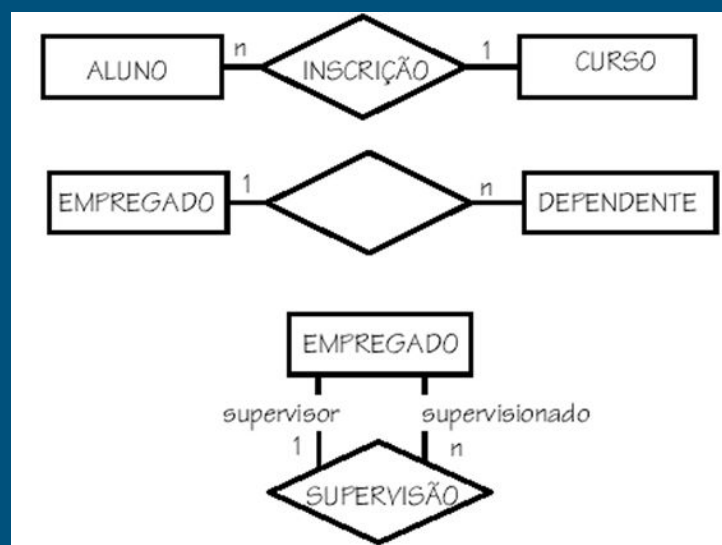
emerson@paduan.pro.br

Relacionamentos 1 : 1



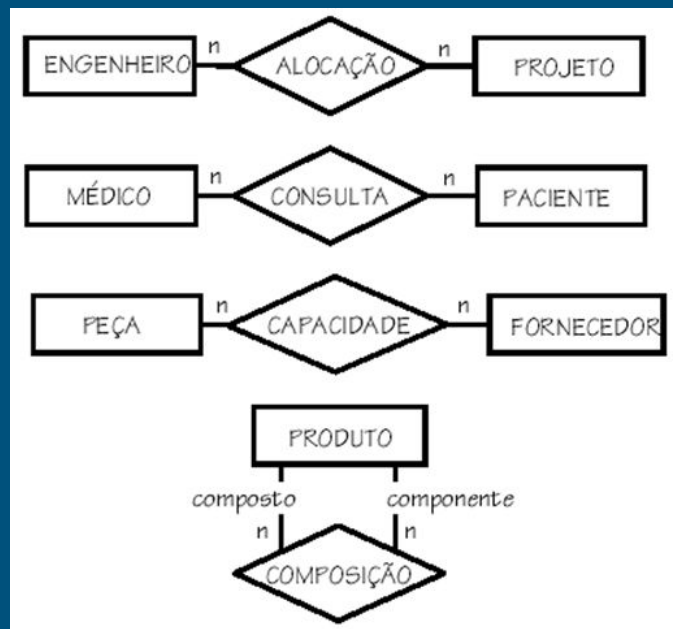
emerson@paduan.pro.br

Relacionamentos 1 : N



emerson@paduan.pro.br

Relacionamentos N : N



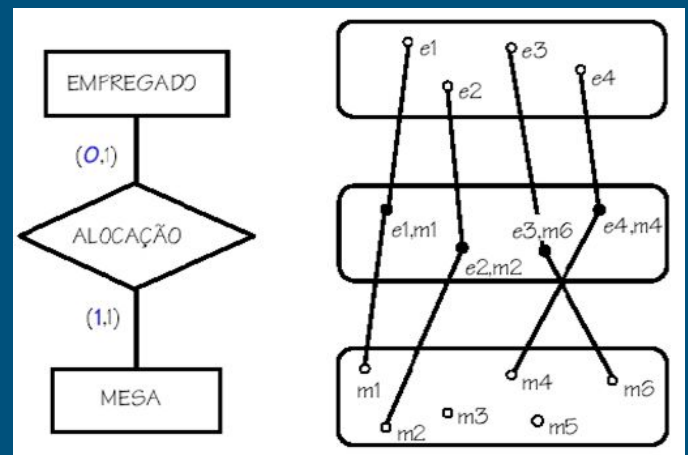
emerson@paduan.pro.br

Cardinalidade mínima

Para fins de projeto de BD, consideram-se apenas duas cardinalidades mínimas:

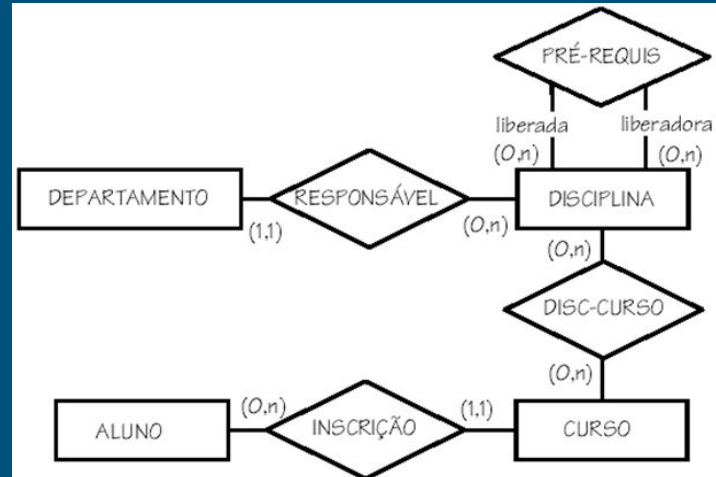
Cardinalidade mínima 0 → Opcional

Cardinalidade mínima 1 → Obrigatória



emerson@paduan.pro.br

Exemplo

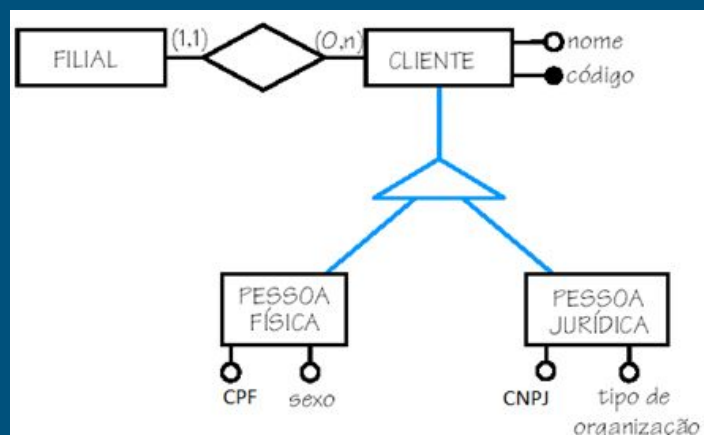


emerson@paduan.pro.br

Generalização/Especialização

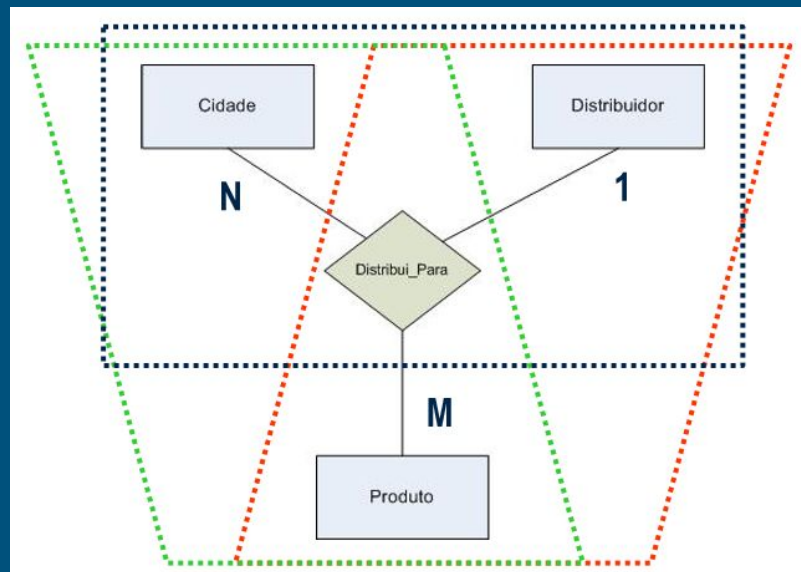
➤ Generalização/Especialização

- Conceito que permite atribuir propriedades particulares a um subconjunto das ocorrências (especializadas) de uma entidade genérica.



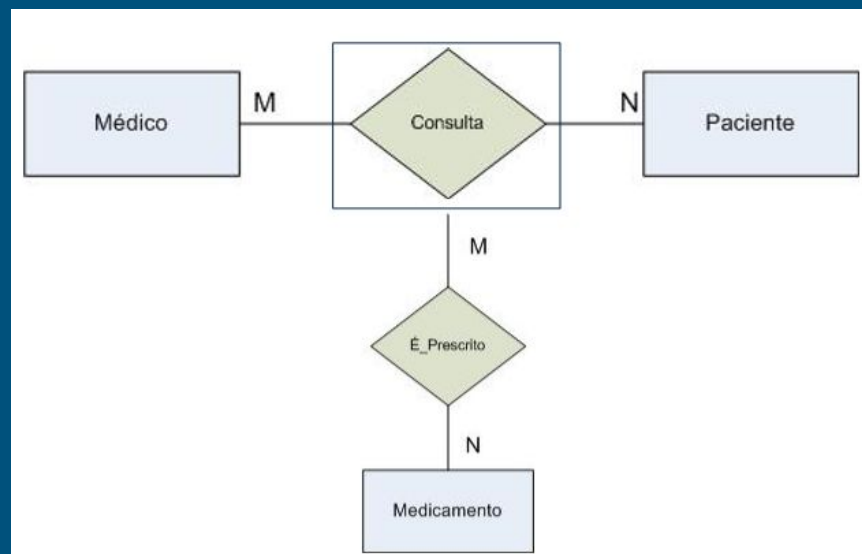
emerson@paduan.pro.br

Relacionamentos ternários



emerson@paduan.pro.br

Entidades Associativas

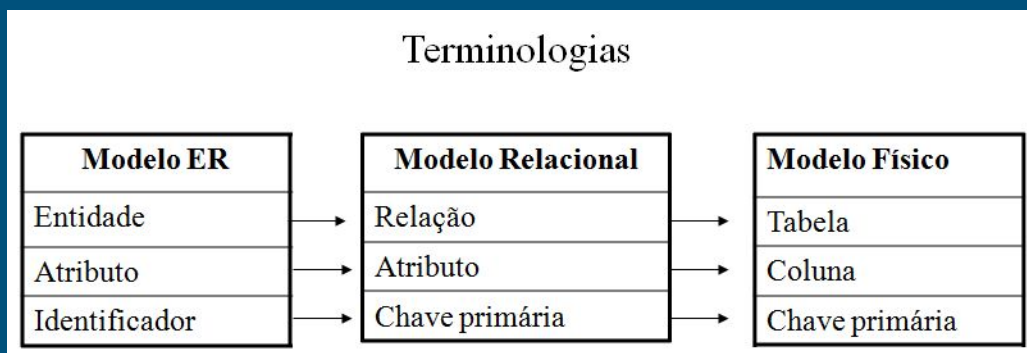


emerson@paduan.pro.br

Mapeamento:

Lógico → Relacional

Terminologias



Terminologias

The diagram shows a table with four columns and five rows. Annotations point to specific parts of the table:

- Coluna (atributo)**: Points to the 'CodigoDeppto' column.
- nome do campo (nome do atributo)**: Points to the 'CategFuncional' column.
- valor do campo (valor do atributo)**: Points to the value 'C2' in the 'CategFuncional' column of the third row.
- linha (tupla)**: Points to the entire third row of the table.

Emp	CódigoEmp	Nome	CodigoDeppto	CategFuncional
	E5	Souza	D1	C5
	E3	Santos	D2	C5
	E2	Silva	D1	C2
	E1	Soares	D1	—

emerson@paduan.pro.br

Transformação MER → Relacional

Regras gerais de transformação

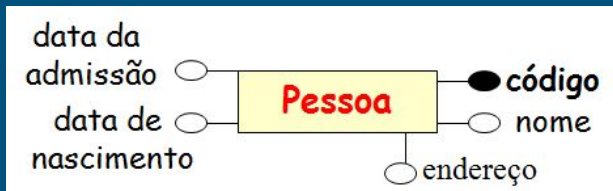
Aplicáveis à maioria dos casos

Há exceções ou situações nas quais, por exigência da aplicação, outros mapeamentos são utilizados.

emerson@paduan.pro.br

Entidades

- Toda entidade gera uma tabela
- Cada atributo de uma entidade corresponde a uma coluna da tabela
- Atributos identificadores correspondem às colunas que compõem a chave-primária da tabela



Pessoa (CodigoPess, Nome, Endereço, DataAdm, DataNasc)

emerson@paduan.pro.br

Chaves

Um chave primária é uma coluna, ou uma combinação de colunas, cujos valores distinguem uma linha das demais dentro de uma tabela.

Uma chave estrangeira é uma coluna, ou uma combinação de colunas, cujos valores referenciam a chave primária de outra tabela. (relacionamentos)

DEPTO	
CodigoDepto	NomeDepto
D1	Compras
D2	Engenharia
D3	Vendas

CodigoDepto em EMP é uma chave estrangeira em relação a tabela DEPTO

EMP				
CodigoEmp	Nome	CodigoDepto	CategFuncional	CIC
E1	Souza	D1	-	132.121.331-20
E2	Santos	D2	C5	891.221.111-11
E3	Silva	D2	C5	341.511.775-45
E5	Soares	D1	C2	631.692.754-88

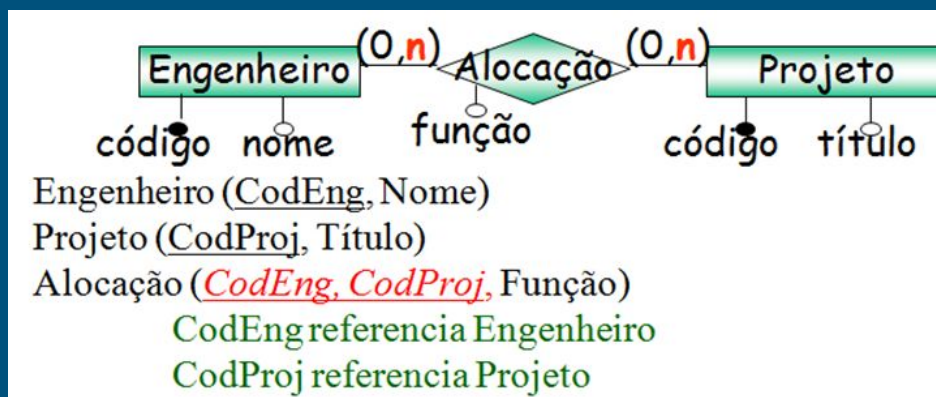
emerson@paduan.pro.br

Relacionamentos

- Tabela própria: O relacionamento é implementado como uma tabela própria contendo as colunas correspondentes aos identificadores das entidades relacionadas.
- Adição de colunas: Adicionar colunas correspondentes aos identificadores da entidade com cardinalidade máxima 1 à outra entidade envolvida no relacionamento.
- Fusão de tabelas: Somente é possível quando o relacionamento é do tipo 1:1.

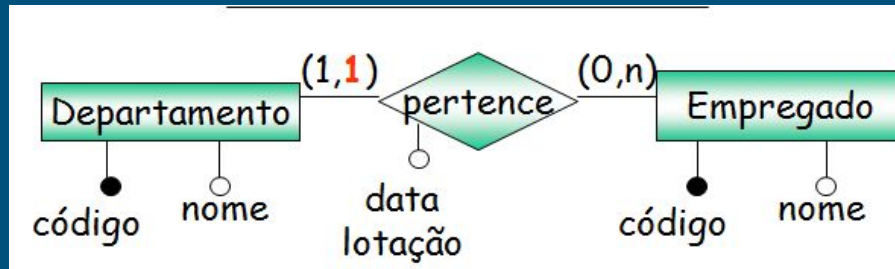
emerson@paduan.pro.br

Relacionamentos n: n



emerson@paduan.pro.br

Relacionamentos 1 : n



Departamento (CodDept, Nome)

Empregado (CodEmp, Nome, CodDept)

CodDept referencia Departamento

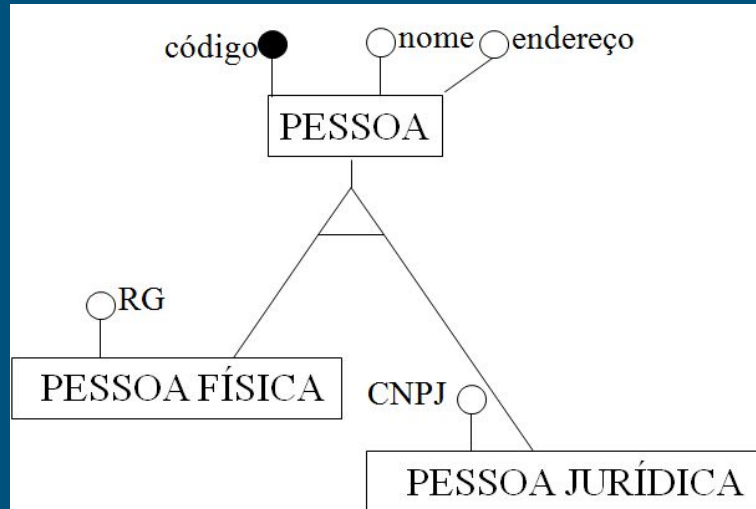
emerson@paduan.pro.br

Relacionamentos 1 : 1

- adição de colunas
- utilização de tabela própria
- fusão de tabelas

emerson@paduan.pro.br

Generalização/Especialização



emerson@paduan.pro.br

Generalização/Especialização

○ Solução 1: Tabela única

Tem que aceitar valores null

Pessoa(codigo, nome, endereco, tipo, RG, CNPJ)

↑
Pessoa Física ou Jurídica

emerson@paduan.pro.br

Generalização/Especialização

- o **Solução 2:** Uma tabela para cada entidade

Pessoa(codigo, nome, endereco)

Pessoa_Fisica(codigo, RG)

codigo referencia Pessoa

Pessoa_Juridica(codigo, CNPJ)

codigo referencia Pessoa

emerson@paduan.pro.br

Mapeamento:

Relacional → Físico

Projeto físico

Para cada uma das tabelas definidas no modelo lógico do banco de dados deve-se:

- definir os tipos de dados de cada coluna;
- definir as restrições de integridade correspondentes.

emerson@paduan.pro.br

Restrições de integridade

- Primary key - (chave primária)
- Foreign key - (chave estrangeira)
- Unique – valores únicos
- Default – assumir valor default
- Check – verificar condição para o domínio coluna

emerson@paduan.pro.br

chaves

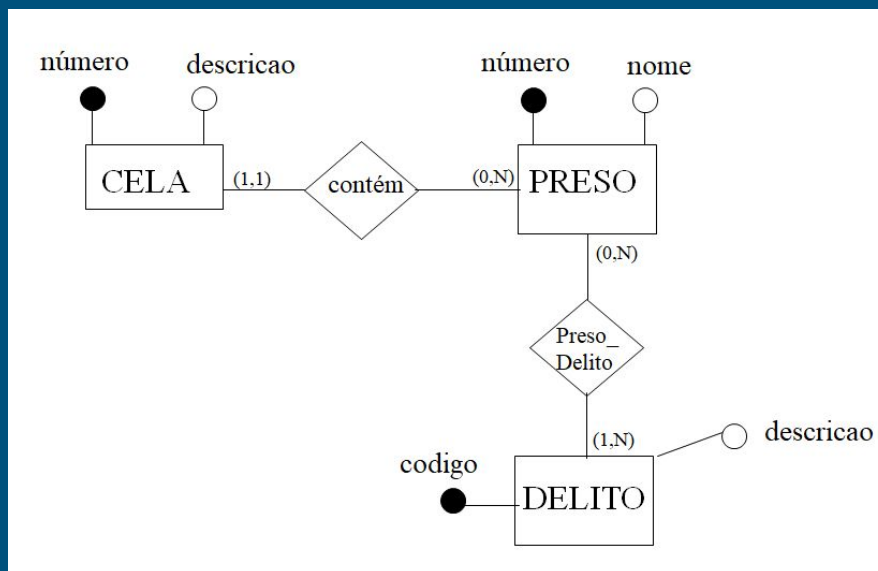
emerson@paduan.pro.br

Modelo

Nome da Tabela	Nome da tabela que será criada no banco de dados			
Descrição	Breve descrição sobre o objetivo da tabela.			
Responsável	Nome do responsável pela manutenção na documentação da tabela.			
Coluna	Descrição	Tipo de dado	Nulo	Consistência
<coluna 1>	Descrição da coluna	Tipo de dado	Aceita Null ? (N = não e S = Sim)	Tipo de Consistência associada ao campo, exemplo: PK, FK, UNIQUE, DEFAULT, CHECK
<coluna 2>	Descrição da coluna	Tipo de dado	Aceita Null? (N = não e S = Sim)	Tipo de Consistência associada ao campo, exemplo: PK, FK, UNIQUE, DEFAULT, CHECK
<coluna ...>	Descrição da coluna	Tipo de dado	Aceita Null? (N = não e S = Sim)	Tipo de Consistência associada ao campo, exemplo: PK, FK, UNIQUE, DEFAULT, CHECK

emerson@paduan.pro.br

Exemplo



emerson@paduan.pro.br

Exemplo

Cela (nrcela, descricao)

Preso(numero, nome, *nrcela*, inicio_pena, sexo, duracao_pena)
nrcela referencia Cela

Delito(codigo, descricao)

Preso_Delito(numero, codigo)
numero referencia Preso
codigo referencia Delito

Onde:

* PK estão grifadas

** FK estão em itálico

emerson@paduan.pro.br

Exemplo

Nome da Tabela	CELA			
Descrição	Estrutura que armazena informações sobre as celas existentes no presídio.			
Responsável	Maria Antônia			
Coluna	Descrição	Tipo de dado	Nulo	Consistência
nr_cela	número da cela	Number(3)	N	PK
decricao	<u>Descrição</u> detalhada da cela em questão	Varchar(30)	N	

emerson@paduan.pro.br

Exemplo

Nome da Tabela	PRESO			
Descrição	Estrutura que armazena informações sobre os presos.			
Responsável	Maria Antônia			
Coluna	Descrição	Tipo de dado	Nulo	Consistência
numero_preso	ID do preso	Number(3)	N	PK
cod_cela	ID da cela	Char(3)	N	FK (CELA)
nome	Nome do preso	Varchar2(30)	N	
inicio_pena	Data de inicio de pena	Date	S	
sexo	Sexo do preso	Char(1)	N	Default ('M') Check → 'M' - <u>masculino</u> 'F' - <u>feminine</u>
duracao_pena	Pena total em meses	Number	S	

emerson@paduan.pro.br

Exemplo

Nome da Tabela	DELITO			
Descrição	Estrutura que armazena informações sobre os tipos de delitos que podem ter sido cometidos pelos presos.			
Responsável	João Paulo			
Coluna	Descrição	Tipo de dado	Nulo	Consistência
codigo	Código do delito	Number(3)	N	PK
decricao	Descrição do delito	Varchar(30)	N	

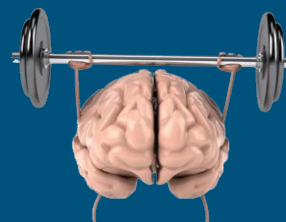
emerson@paduan.pro.br

Exemplo

Nome da Tabela	PRESO_DELITO			
Descrição	Estrutura que armazena informações quais delitos forma cometidos pelos presos.			
Responsável	João Paulo			
Coluna	Descrição	Tipo de dado	Nulo	Consistência
codigo	Código do delito	Number(3)	N	PK e FK(DELITO)
numero	Número do preso que cometeu o delito em questã	Number(3)	N	PK e FK (PRESO)

emerson@paduan.pro.br

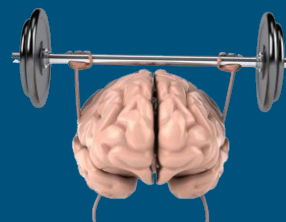
Exercício



FUNCIONÁRIO	MATRÍCULA	DATA NASC.	DATA ADMISSÃO	DEPARTAMENTO	SALÁRIO	ENDEREÇO
Ana Maria	1234	10/02/1986	09/08/2010	01 - Financeiro	R\$ 1.234,56	R. Dario Pereira, 23
Rosa da Silva	1235	15/08/1970	07/06/2005	02 - Marketing	R\$ 2.345,00	Trav. Das Lágrimas, 34
Antonia Camargo	3245	18/07/1986	11/02/2003	01 - Financeiro	R\$ 7.654,99	Pç. XV de novembro, 219
João da Silva	2233		08/09/2011	03 - Recursos Humanos	R\$ 3.452,12	Av. Nações Unidas, 10200
Joaquim Xavier	7866	21/04/1987	05/04/2010	03 - Recursos Humanos	R\$ 980,00	R. Gil Vicente, 45
José Maria	9876	08/09/1998	17/03/2009	04 - Comercial	R\$ 1.780,00	Av. Paulista, 23
FUNCIONÁRIO	DEPENDENTE	DATA NASC.		PROJETO	DATA INÍCIO	DATA TÉRMINO
Rosa da Silva	Pedro da Silva	12/01/1963		001 - PROJETO ABC	10/01/2013	
Antonia Camargo	George Camargo	15/09/1971		002 - PROJETO BCD	28/10/2012	10/03/2013
Antonia Camargo	Antonio Camargo	10/06/2000		003 - PROJETO CDE	09/12/2012	
João da Silva	Ana da Silva	23/04/1976		004 - PROJETO DEF	15/12/2012	
João da Silva	Joana da Silva	16/05/2002				
João da Silva	João da Silva Jr.	16/05/2002				
PROJETO	FUNCIONÁRIO	DATA ENTRADA	DATA SAÍDA			
001 - PROJETO ABC	Antonia Camargo	10/01/2013				
002 - PROJETO BCD	Antonia Camargo	28/10/2012	10/03/2013			
003 - PROJETO CDE	João da Silva	09/12/2012				
003 - PROJETO CDE	Joaquim Xavier	15/01/2013				

emerson@paduan.pro.br

Exercício

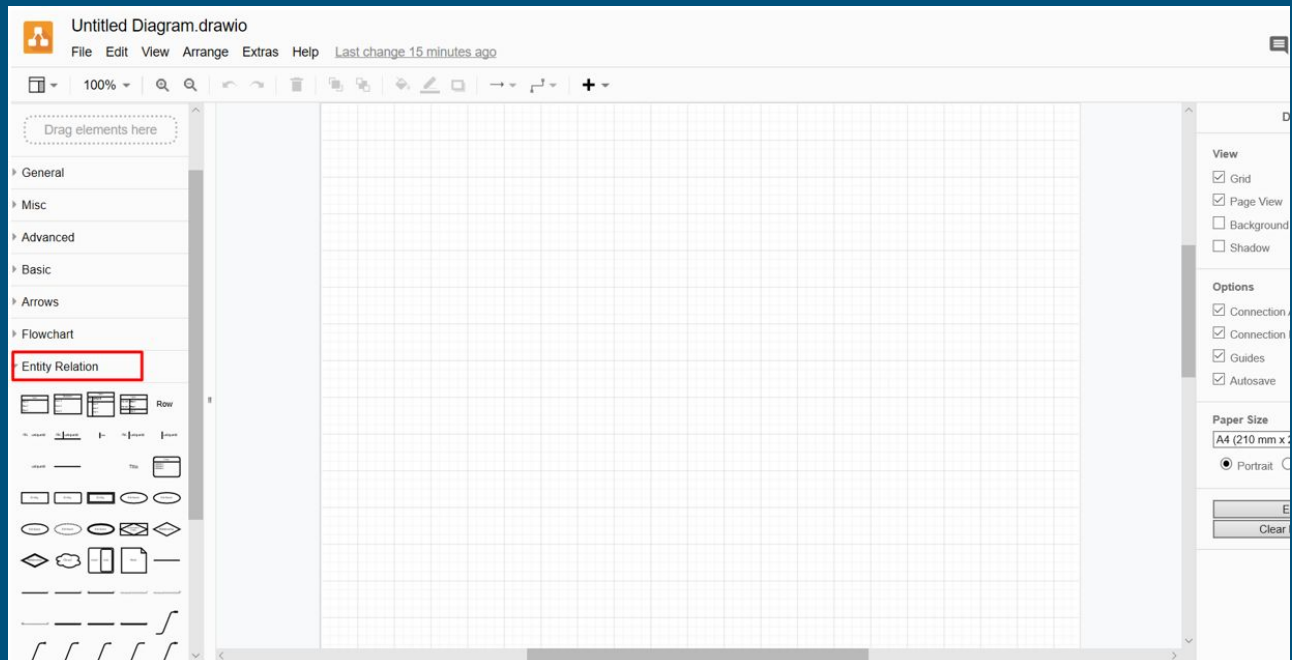


Nossa empresa necessita de um catálogo de CD's, pois constantemente recebemos ligações ou visitas a nossa loja. Os clientes costumam perguntar o nome, o preço de venda e a gravadora do CD. Cada CD contém diversas faixas em que ficam gravadas as músicas. Cada música tem seu tempo de duração e é importante sabermos os autores delas. Além disso, é importante conhecermos outras informações da gravadora, visto que, na falta de um CD, podemos localizar mais facilmente o local onde efetuar a compra. Classificamos os CD's com base na faixa de preços a que ele pertence.

Neste momento ainda não precisamos registrar os intérpretes das músicas, bem como as diferentes gravações que uma música pode ter.

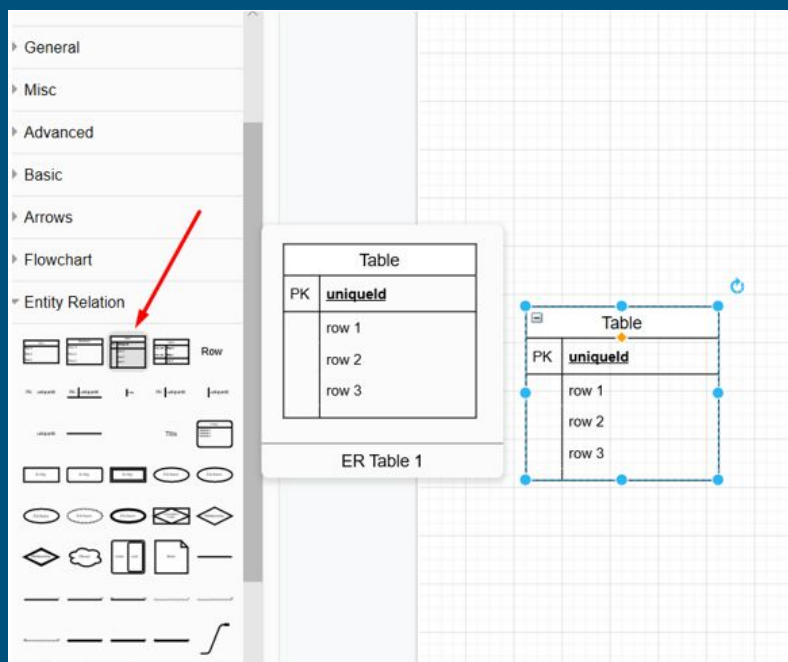
emerson@paduan.pro.br

Exercício



emerson@paduan.pro.br

Exercício



emerson@paduan.pro.br

Exercício

FUNCIONARIOS	
PK	<u>MATRICULA (INT)</u>
	NOME (VARCHAR(40))
	SALARIO (DECIMAL(8,2))
	ENDERECO (VARCHAR(80))
	DATAADMISSAO (DATE)
	DATANASCIMENTO (DATE)
	DEPARTAMENTO (VARCHAR (30))

DEPENDENTES	
PK	<u>CODIGO (INT)</u>
	NOME (VARCHAR(60))
	DATANASCIMENTO (DATE)
FK	MATRICULA (INT)

PROJETOS	
PK	<u>CODIGO (INT)</u>
	DESCRICAO (VARCHAR(60))
	DATAINICIO (DATE)
	DATATERMINO (DATE)

HISTORICOPROJETOS	
PK,FK1	<u>MATRICULA (INT)</u>
PK,FK2	<u>CODIGO_PROJETO (INT)</u>
	DATAENTRADA (DATE)
	DATASAIIDA (DATE)

emerson@paduan.pro.br

MySQL



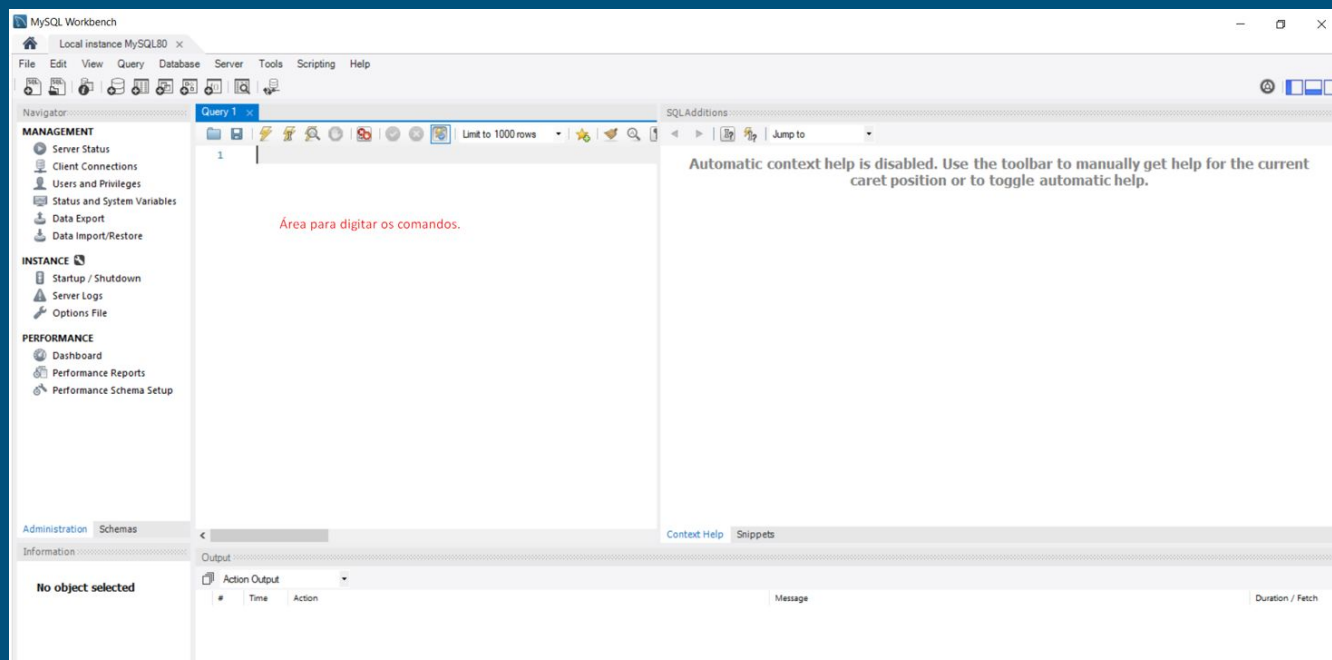
emerson@paduan.pro.br

Tipos de dados

Tipo de Dados	Descrição
CHAR	Conjunto de caracteres com tamanho fixo.
VARCHAR	Conjunto de caracteres com tamanho variável.
DECIMAL(p,s)	Um número com precisão (p) e escala(s). A precisão varia de 1 a 38 e a escala pode ser -84 a 127.
DATE	Valores de data de 1º janeiro de 4712 A.C. a 31 dezembro de 9999 A.D.
BLOB	Conjunto de binários (utilizado para armazenar imagens).
CLOB	Conjunto de caracteres (utilizado para armazenar arquivos).
TIMESTAMP	Ano, mês, dia, hora, minuto, segundo e segundos fracionários. O valor de segundos fracionários pode variar de 0 a 9 (precisão até um bilionésimo de segundo). O padrão é 6 (um milionésimo).

emerson@paduan.pro.br

MySQL Workbench



emerson@paduan.pro.br

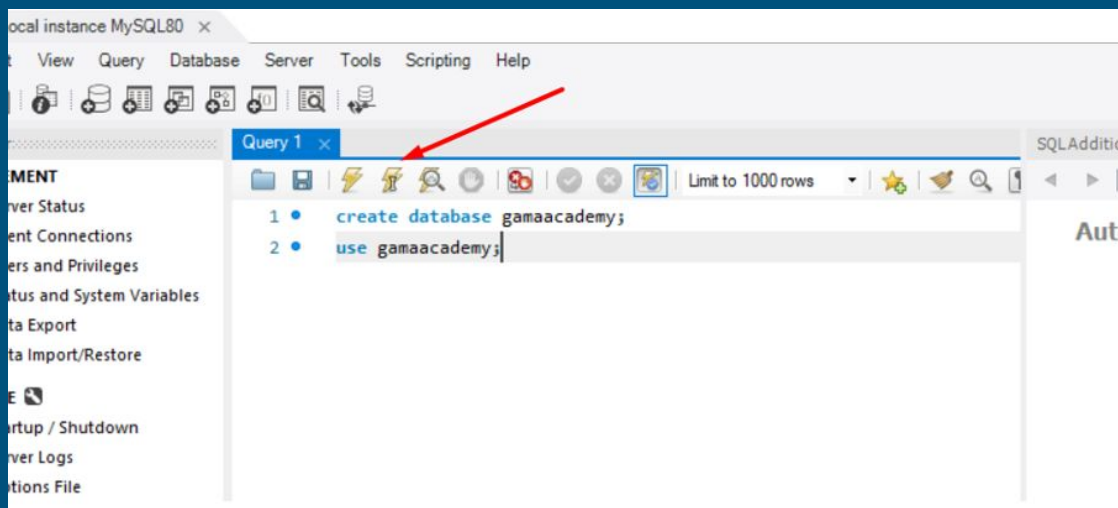
SQL Language

DDL - Data Definition Language

DML - Data Manipulation Language

emerson@paduan.pro.br

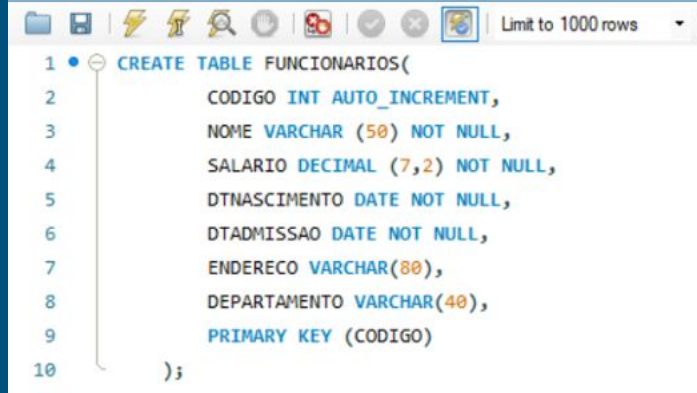
SQL Language



emerson@paduan.pro.br

SQL Language

CREATE TABLE <nome-tabela> (<nome-coluna> <tipo-do-dado> [NOT NULL]
PRIMARY KEY (nome-coluna-chave) FOREIGN KEY
(nome-coluna-chave-estrangeira) REFERENCES <nome-tabela-pai>
(nome-coluna-chave-primária));

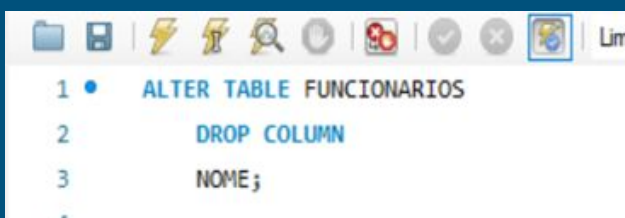


```
1 CREATE TABLE FUNCIONARIOS(  
2     CODIGO INT AUTO_INCREMENT,  
3     NOME VARCHAR (50) NOT NULL,  
4     SALARIO DECIMAL (7,2) NOT NULL,  
5     DTNASCIMENTO DATE NOT NULL,  
6     DTADMISSAO DATE NOT NULL,  
7     ENDERECO VARCHAR(80),  
8     DEPARTAMENTO VARCHAR(40),  
9     PRIMARY KEY (CODIGO)  
10 );
```

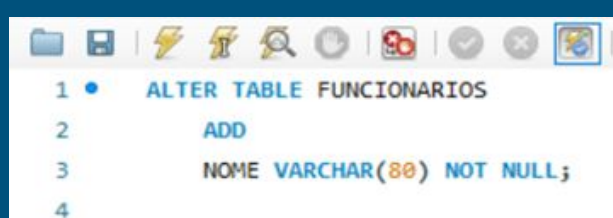
emerson@paduan.pro.br

SQL Language

ALTER TABLE <nome-tabela>
DROP COLUMN <nome-coluna>
ADD <nome-coluna> <tipo-do-dado> [NOT NULL] [NOT NULL WITH DEFAULT] RENAME
<nome-coluna> <novo-nome-coluna>
MODIFY <nome-coluna> <tipo-do-dado> [NULL] [NOT NULL]



```
1 ALTER TABLE FUNCIONARIOS  
2 DROP COLUMN  
3 NOME;
```



```
1 ALTER TABLE FUNCIONARIOS  
2 ADD  
3 NOME VARCHAR(80) NOT NULL;  
4
```

emerson@paduan.pro.br

SQL Language

DROP TABLE

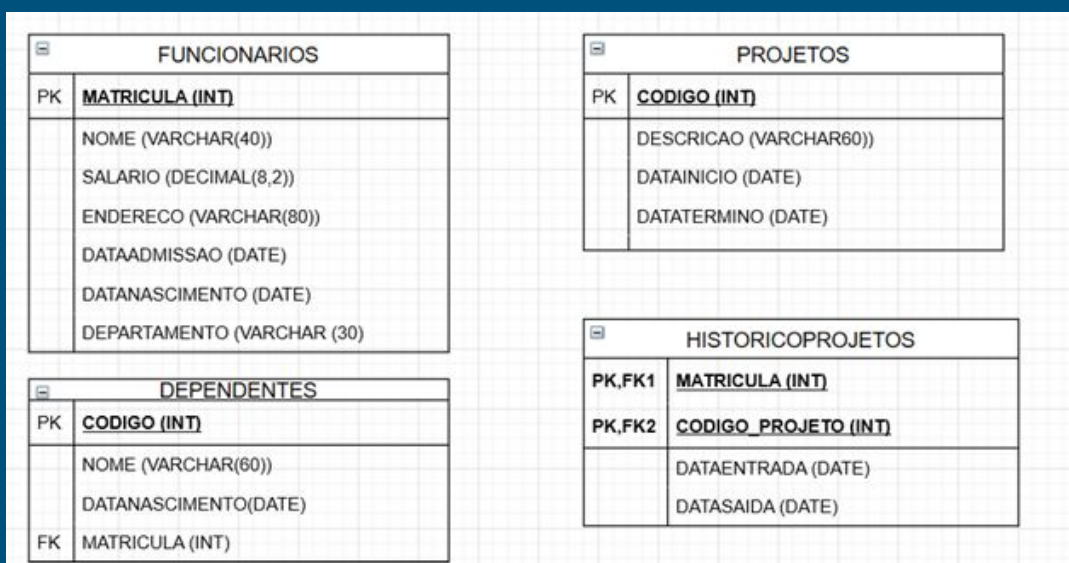
Deletar a estrutura e os dados existentes em uma tabela. Após a execução deste comando estarão deletados todos os dados, estrutura e índices de acessos que estejam a ela associados.

Sintaxe:

DROP TABLE <nome-tabela>;

emerson@paduan.pro.br

Exercício



emerson@paduan.pro.br

Exercício

```
1 CREATE TABLE FUNCIONARIOS(  
2     CODIGO INT AUTO_INCREMENT,  
3     NOME VARCHAR (50) NOT NULL,  
4     SALARIO DECIMAL (7,2) NOT NULL,  
5     DTNASCIMENTO DATE NOT NULL,  
6     DTADMISSAO DATE NOT NULL,  
7     ENDereco VARCHAR(80),  
8     DEPARTAMENTO VARCHAR(40),  
9     PRIMARY KEY (CODIGO)  
10 );
```

```
1 CREATE TABLE DEPENDENTES(  
2     CODIGO INT AUTO_INCREMENT,  
3     NOME VARCHAR (50) NOT NULL,  
4     DTNASCIMENTO DATE NOT NULL,  
5     CODFUNCIONARIO INT,  
6     PRIMARY KEY (CODIGO),  
7     FOREIGN KEY (CODFUNCIONARIO)  
8     REFERENCES FUNCIONARIOS(CODIGO)  
9 );
```

```
Query 1  
1 CREATE TABLE PROJETOS(  
2     CODIGO INT AUTO_INCREMENT,  
3     DESCRICAO VARCHAR (50) NOT NULL,  
4     DTINICIO DATE NOT NULL,  
5     DTERMINO DATE,  
6     PRIMARY KEY (CODIGO)  
7 );
```

```
1 CREATE TABLE HISTORICOPROJETOS(  
2     MATRICULA INT NOT NULL,  
3     CODIGO_PROJETO INT NOT NULL,  
4     DTENTRADA DATE NOT NULL,  
5     DTSaida DATE,  
6     PRIMARY KEY (MATRICULA, CODIGO_PROJETO),  
7     FOREIGN KEY (MATRICULA)  
8     REFERENCES FUNCIONARIOS(CODIGO),  
9     FOREIGN KEY (CODIGO_PROJETO)  
10    REFERENCES PROJETOS(CODIGO)  
11 );
```

emerson@paduan.pro.br

SQL Language

INSERT INTO <nome-tabela> [(<nome-coluna>, [<nome-coluna>],...)] VALUES (<conteúdo>, [<conteúdo>], ...);

```
1 INSERT INTO FUNCIONARIOS  
2 (NOME, SALARIO, DTNASCIMENTO, DTADMISSAO, ENDereco, DEPARTAMENTO)  
3 VALUES  
4 ("XPTO", 5000, str_to_date("17/06/1975", "%d/%m/%Y"), str_to_date("01/05/2020", "%d/%m/%Y"), "RUA X", "TI");
```

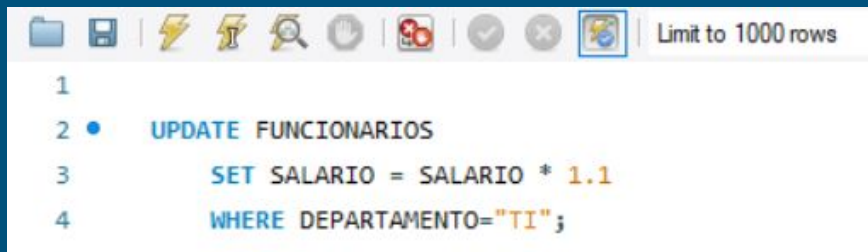
* Para zerar o auto incremento:

ALTER TABLE <TABELA> AUTO_INCREMENT=1

emerson@paduan.pro.br

SQL Language

UPDATE <nome-tabela> SET <nome-coluna> = <novo conteúdo para o campo>
[,<nome-coluna> = <novo conteúdo para o campo>] [WHERE <condição>]

A screenshot of a SQL query editor window. The toolbar at the top includes icons for file operations, execution, and a 'Limit to 1000 rows' option. The query text is as follows:

```
1  
2 • UPDATE FUNCIONARIOS  
3     SET SALARIO = SALARIO * 1.1  
4     WHERE DEPARTAMENTO="TI";
```

emerson@paduan.pro.br

SQL Language

DELETE FROM <nome-tabela> [WHERE <condição>]

A screenshot of a SQL query editor window. The toolbar at the top includes icons for file operations, execution, and a 'Limit to 1000' option. The query text is as follows:

```
1 • DELETE FROM FUNCIONARIOS  
2     WHERE CODIGO=1;
```

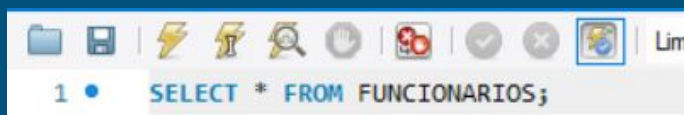
emerson@paduan.pro.br

SQL Language

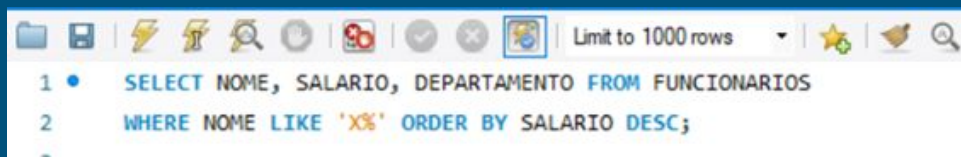
```
SELECT tabela1.coluna, tabela2.coluna  
FROM tabela1  
[CROSS JOIN tabela2] |  
[NATURAL JOIN tabela2] |  
[JOIN tabela2 USING (nome_coluna)] |  
[JOIN tabela2  
ON(tabela1.nome_coluna = tabela2.nome_coluna)] |  
[LEFT|RIGHT|FULL OUTER JOIN tabela2  
ON (tabela1.nome_coluna = tabela2.nome_coluna)];
```

emerson@paduan.pro.br

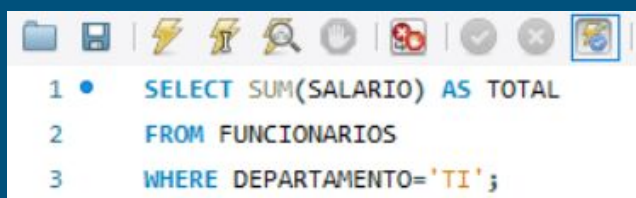
SQL Language



1 • `SELECT * FROM FUNCIONARIOS;`



1 • `SELECT NOME, SALARIO, DEPARTAMENTO FROM FUNCIONARIOS`
2 `WHERE NOME LIKE 'X%' ORDER BY SALARIO DESC;`
3



1 • `SELECT SUM(SALARIO) AS TOTAL`
2 `FROM FUNCIONARIOS`
3 `WHERE DEPARTAMENTO='TI';`

emerson@paduan.pro.br

SQL Language

SELECT

Pesquisas em múltiplas tabelas

Se faz necessário quando precisamos recuperar informações que estão distribuídas em uma ou várias tabelas, a esta necessidade damos o nome de JUNÇÕES(JOIN).

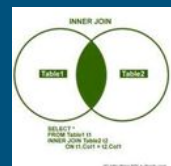
A operação JUNÇÃO (JOIN) é usada para combinar registros relacionados, dentro de um único registro. Esta operação nos permite processar os relacionamentos entre as relações.

emerson@paduan.pro.br

SQL Language

SELECT – INNER JOIN

Pesquisas em múltiplas tabelas

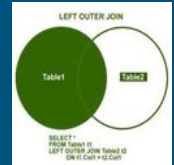


Nesse tipo de junção, caso sejam unidas duas tabelas serão exibidos todos os dados existentes nas duas tabelas envolvidas na consulta.

```
Limit to 1000 rows
1 • INSERT INTO DEPENDENTES VALUES (1, 'FILHO1', str_to_date('26/11/1998', '%d/%m/%Y'), 2);
2 • INSERT INTO DEPENDENTES VALUES (2, 'FILHO2', str_to_date('23/01/2020', '%d/%m/%Y'), 2);
3 • INSERT INTO DEPENDENTES VALUES (3, 'FILHO1 - OUTRO', str_to_date('20/04/2017', '%d/%m/%Y'), 3);
4
5 • SELECT D.NOME AS FILHO,
6       date_format(D.DTNASCIMENTO, '%d/%m/%Y') AS NASCIMENTO,
7       F.NOME AS COLABORADOR
8       FROM FUNCIONARIOS AS F
9       INNER JOIN DEPENDENTES AS D
10      ON F.CODIGO=D.CODFUNCIONARIO
11     WHERE F.NOME='XPTO';
```

emerson@paduan.pro.br

SQL Language



SELECT – LEFT JOIN

Todas as linhas da tabela à esquerda serão recuperadas, independentemente da existência de ocorrências relacionadas na tabela da direita. Preserva as linhas sem correspondência da primeira tabela (esquerda), juntando-as com a linha nula da segunda tabela(direita).

```
1 • INSERT INTO FUNCIONARIOS
2   (NOME, SALARIO, DTNASCIMENTO, DTADMISSAO, ENDereco, DEPARTAMENTO)
3   VALUES
4   ("REGINA", 30000, str_to_date("18/10/1992", "%d/%m/%Y"), str_to_date("04/05/2020", "%d/%m/%Y"), "AVENIDA PAULISTA", "TI");
5
6 • SELECT D.NOME AS FILHO,
7        date_format(D.DTNASCIMENTO, '%d/%m/%Y') AS NASCIMENTO,
8        F.NOME AS COLABORADOR
9   FROM FUNCIONARIOS AS F
10  LEFT JOIN DEPENDENTES AS D
11    ON F.CODIGO=D.CODFUNCIONARIO;
```

emerson@paduan.pro.br

SQL Language

Resumindo os join's



emerson@paduan.pro.br